Digital Signal Processing Fundamentals (5ESC0)

Filter structure

Elisabetta Peri, Sveta Zinger, Piet Sommen

(e.peri@tue.nl)

Lecture content

We will consider implementation of discrete time systems

We have learnt how to describe these systems in the time domain, frequency domain, Z-domain

Now we will see the meaning of these descriptions when we implement discrete time systems

FIR filter

Let's consider FIR (Finite Impulse Response) filter:

it consists of an impulse response which is finite

System function and Difference Equation causal FIR:

$$H(z) = \sum_{n=0}^{N} h[n]z^{-n} \quad \Rightarrow \quad y[n] = \sum_{k=0}^{N} h[k]x[n-k]$$

***Notes*:**

- Requires $N+1$ multiplications and N additions
- Impulse response: $h[n] = h[0]\delta[n] + \cdots h[N]\delta[n-N]$

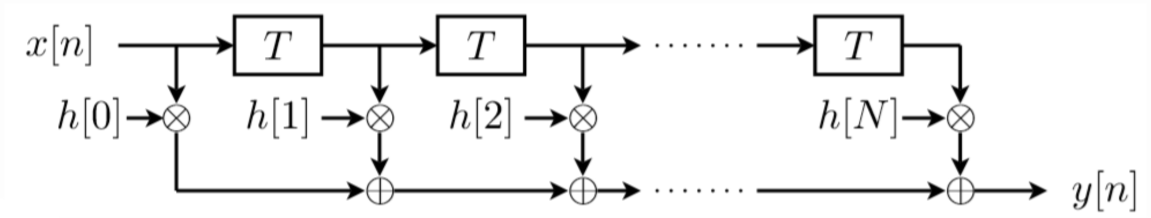
FIR filter

- $H(z)$ contains only zeros

- **Zeros**: The zeros of $H(z)$ are the values of z that yields $H(z) = 0$.
- **Poles**: Poles are values of z that cause $H(z)$ go to infinity. For FIR filters, the denominator is simply 1. Hence there are no poles except at $z = \infty$ and $z = 0$.
- FIR filters do not have feedback loops. Feedback would introduce additional poles (other than at $z = \infty$).

FIR filter structure

Direct form: (*Alternative names*: Transversal filter, Tapped delay line)



The figure shows that we have

- an input,
- a delay line,
- and we multiply each delayed value of a system by $h[n]$,
- then the results are added and constitute the output

This system is described by a polynomial expression

FIR filter structure

We can represent a polynomial in several different ways and therefore we can also implement a system in different ways

For example, this polynomial (cascade form, first-order factors), which is a finite-length polynomial, can be written as a product of first order polynomials:

Cascade form:

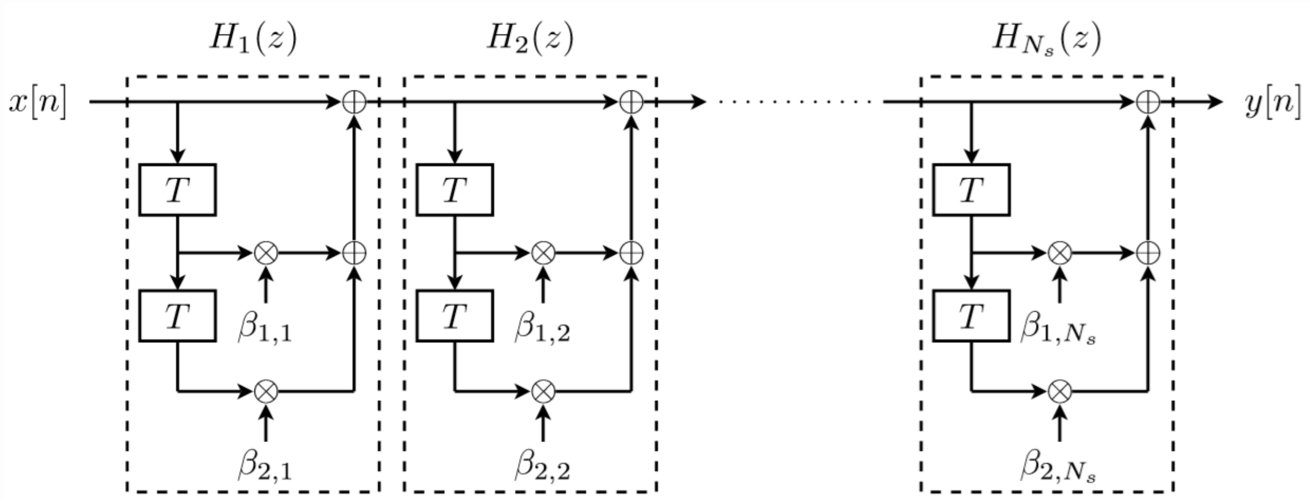
First-order factors : $H(z) = \sum_{n=0}^{N} h[n] z^{-n} = C \prod_{k=1}^{N} (1 - \alpha_k z^{-1})$

Depending on an application the choice can be made between the transversal and cascade form

FIR filter structure

We can also write this as a product of second order polynomials: each second order section contains then two delays and two multiplications

$$H(z) = C \prod_{k=1}^{N_s} \left(1 + \beta_{1,k} z^{-1} + \beta_{2,k} z^{-2}\right)$$



Linear phase filters

A linear phase filter is defined as:

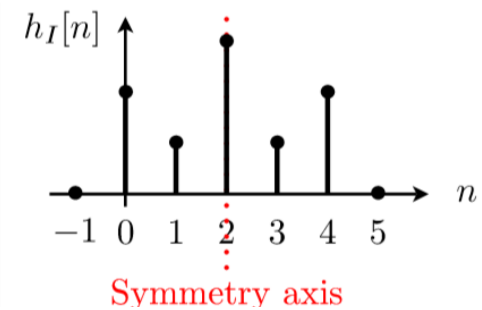
$$H(\omega) = A(\omega)e^{-j(\alpha\omega-\beta)}, \qquad -\pi \leq \omega \leq \pi$$

with a real $A(\omega)$.

- It must satisfy the symmetry property:

  $$h[n] = \epsilon h[N-n]$$

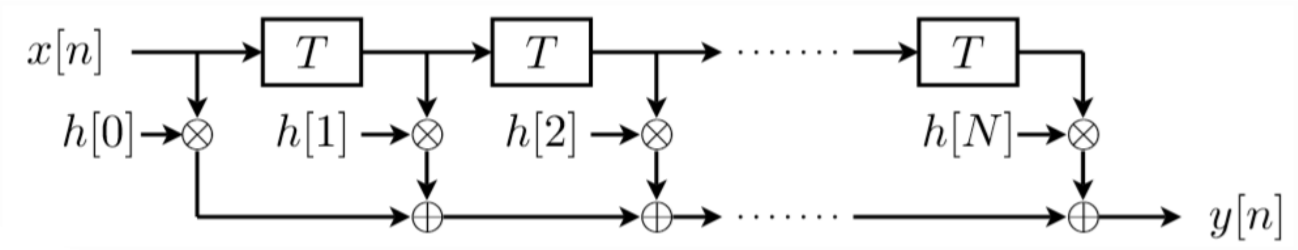
  where $\epsilon = \pm 1$ and N is the filter order.



- The center of the impulse response is at $\frac{N}{2}$ and defines the location of the symmetry axis.
- For odd N, the symmetry axis coincides with one of the sample of $h[n]$, for even N the symmetry axis is between samples of $h[n]$

Linear phase filters

- **Linear phase filter keeps the shape of the input signal** (in the pass band region)**.**
- Phase of each frequency component is linearly dependent on its frequency: $\varphi = \alpha\omega - \beta$
- All samples of the input are delayed by the same amount when passing through the filter.
- The only component that changes at the output is the phase (signal is delayed and such delay is determined by the filter order N).



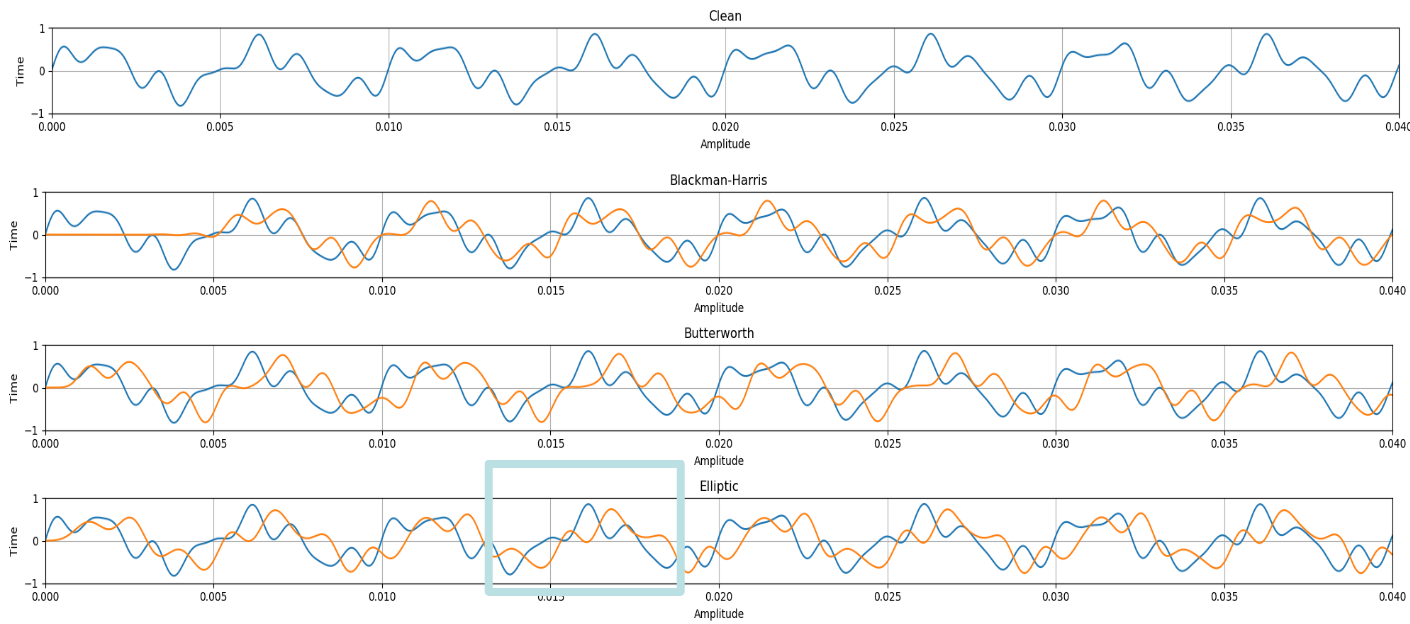
Linear phase filters

- Blue: original signal built as sum of 3 sinusoids
- Orange: filtered version of the original signal passing all the frequency components in the original signal

Linear phase FIR

Quasi-linear phase IIR

IIR



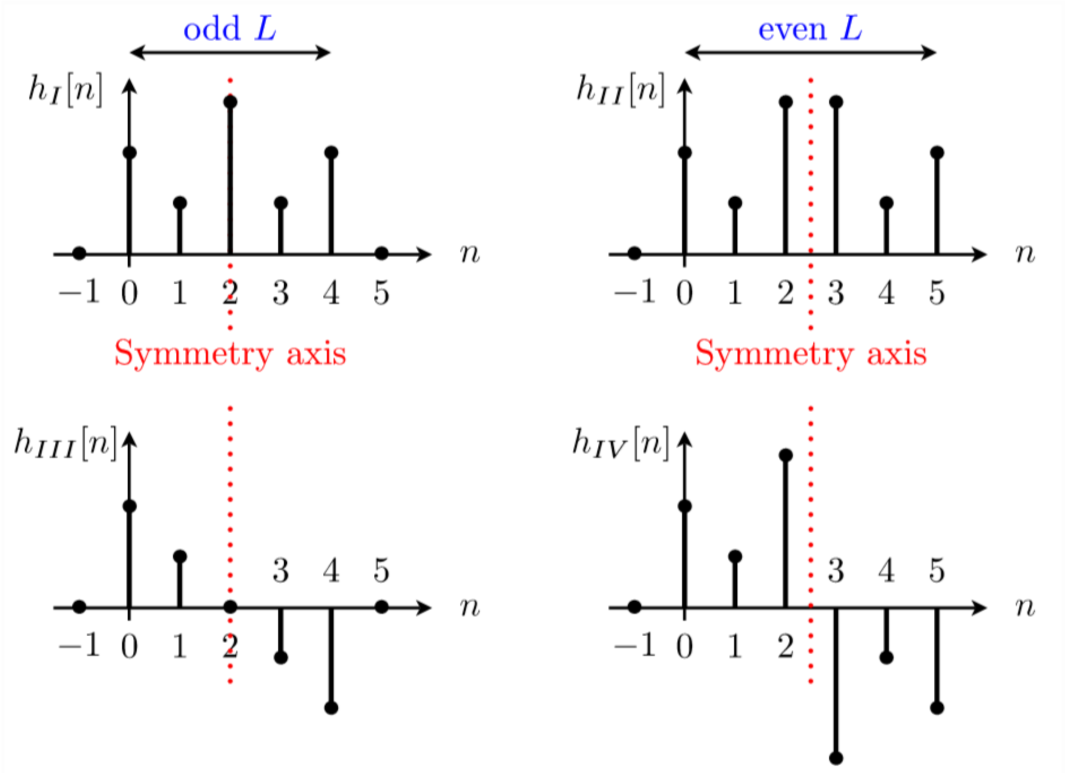
Linear phase filters

- There are 4 different types of linear phase filters.
- Linear phase can only be obtained when the filter is Finite Impulse Response (FIR), and not IIR.
- Impulse response of linear phase FIR has the symmetric or antisymmetric property.

Linear phase filters

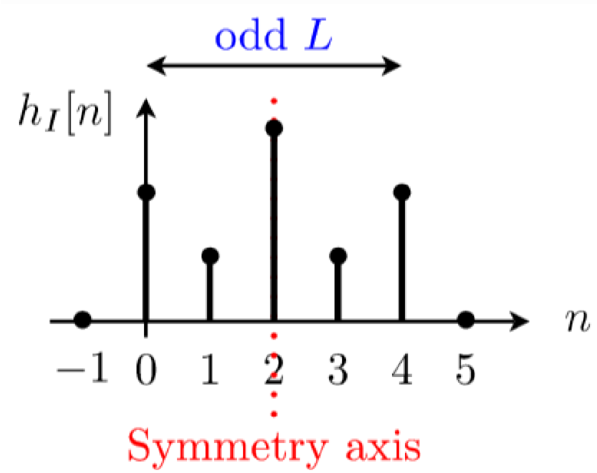
For odd L the symmetry axis coincides with a sample of $h[n]$

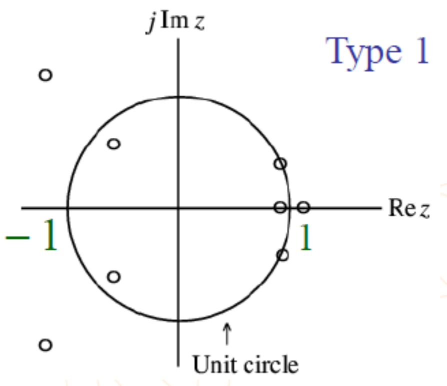
For even L the symmetry axis is between the samples of $h[n]$

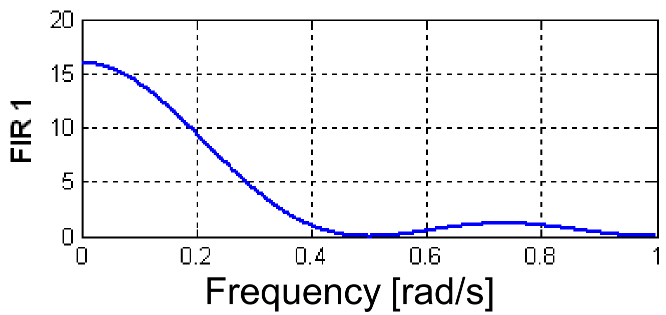


Linear phase filters – type I

- Type I linear phase filter consists of symmetric sequence of **odd** length.
- This filter type has either an even amount of zeros or no zeros at $z = \pm 1$ (i.e. $\vartheta = 0, \pm\pi$). → No Restrictions

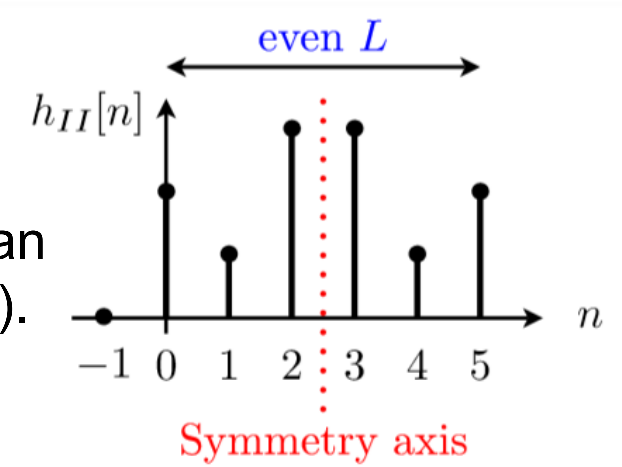


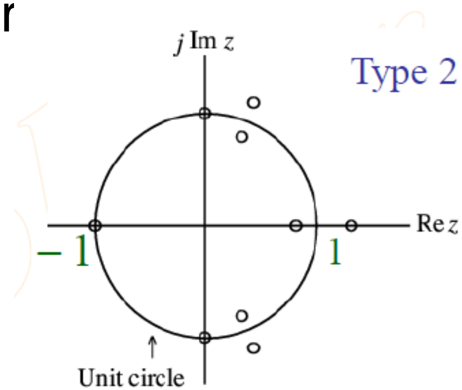


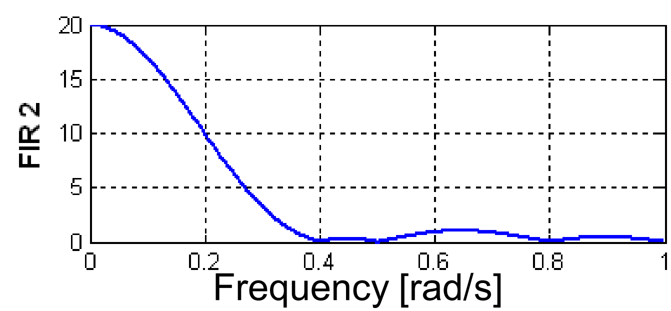


Linear phase filters – type II

- Type II linear phase filter is consists of symmetric sequence of **even** length.
- This filter type has either even amount of zeros or no zeros at $z = 1$ (i.e. $\vartheta = 0$), and an odd number of zeros at $z = -1$ (i.e. $\vartheta = \pm\pi$).
- This filter cannot be used as a high-pass filter

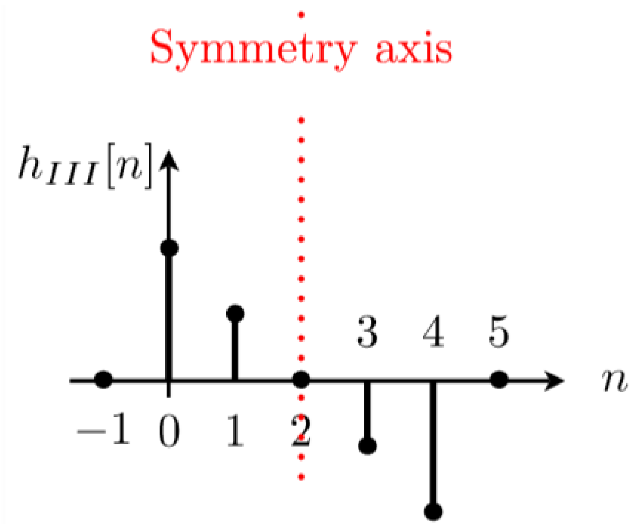


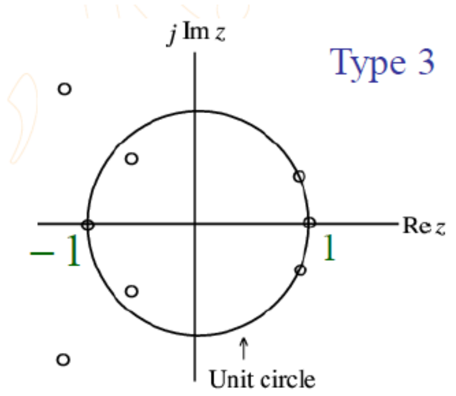


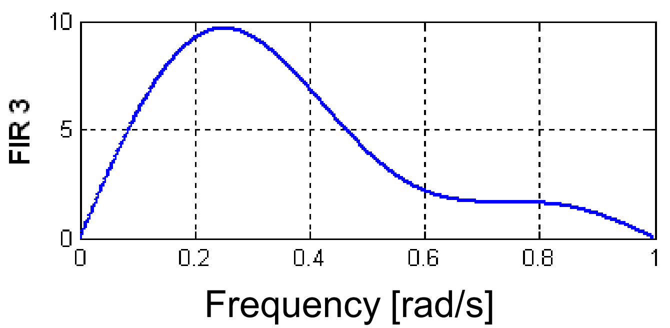


Linear phase filters – type III

- Type III linear phase filter consists of antisymmetric sequence of **odd** length.
- This filter type has an odd amount of zeros at $z = \pm 1$ (i.e. $\vartheta = 0$ and $\vartheta = \pm\pi$).
- This filter cannot be a low-pass or a high-pass filter.
- Can be used as a band-pass filter.

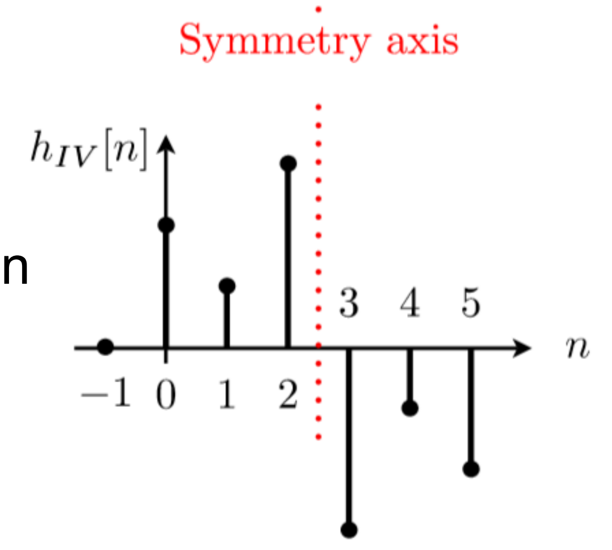


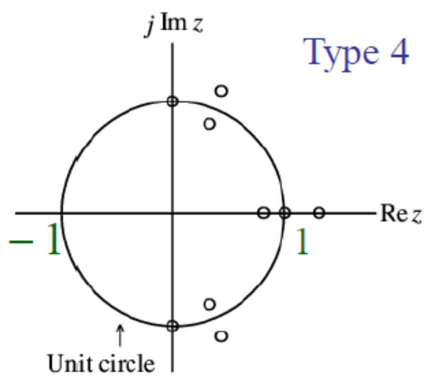


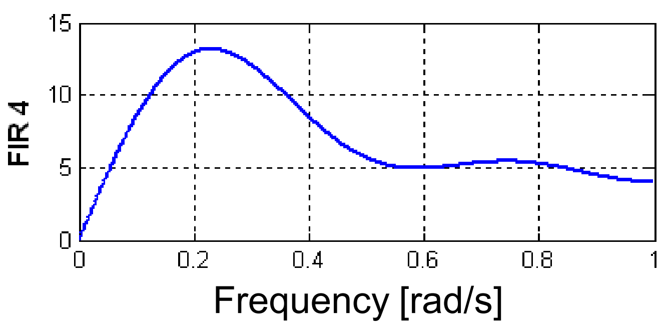


Linear phase filters – type IV

- Type IV linear phase filter consists of antisymmetric sequence of **even** length.
- This filter type has either odd amount of zeros or no zeros at $z = 1$, and either even number of zeros or no zeros at $z = -1$.
- This filter cannot be used as a low-pass filter.







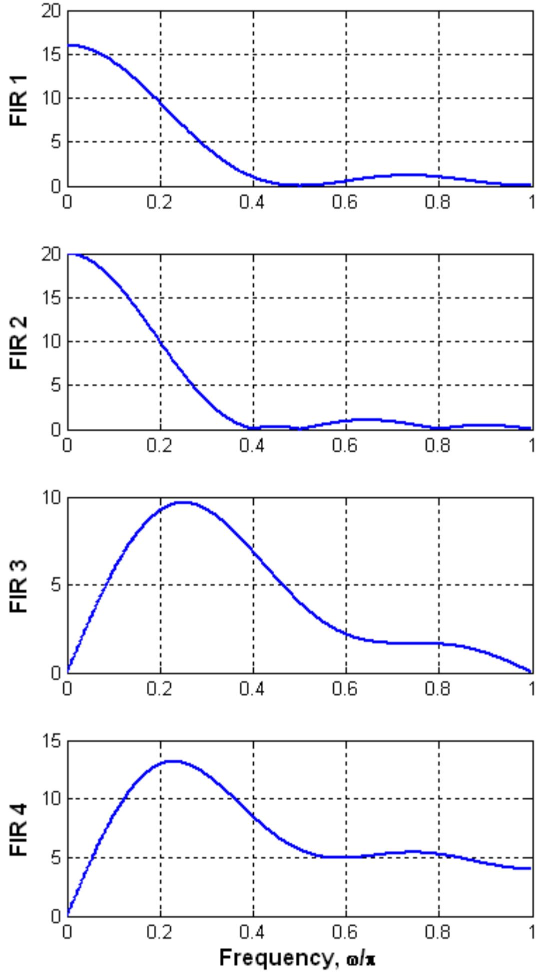
Linear phase filters – Frequency response

Type I FIR filter has no such restrictions and can be used to design almost any type of filter.

A Type II FIR filter cannot be used to design a high-pass filter since it always has a zero at $z = -1$.

A Type III FIR filter has zeros at both $z = \pm 1$, and hence cannot be used to design either a low-pass or a high-pass or a band-stop filter.

A Type IV FIR filter is not appropriate to design a low-pass filter due to the presence of a zero at $z = 1$.



Summary

We considered structures of various filters,

the ways to represent these structures

and to interpret these representations,

and examples